UCRL-84640
PREPRINT

DASLL - AN AUTOMATIC PRINTED CIRCUIT BOARD LAYOUT SYSTEM

W. G. Magnuson, Jr.

This paper was prepared for presentation
at the 1980 IEEE International Conference
on Circuits and Computers (ICCC 80)
New York, NY

October 1-3, 1980

DASLL - AN AUTOMATIC PRINTED CIRCUIT BOARD LAYOUT SYSTEM

W. G. Magnuson, Jr.
Electronics Engineering Department
Lawrence Livermore National Laboratory
Livermore, CA 94550

ABSTRACT

DASLL stands for Design Automation System at Lawrence Livermore and is a system of computer programs to automatically lay out printed circuit boards. The focus has been on two-sided PCB fabrication aids; primarily drill tape, documentation, and artwork generation (including etch, silkscreen, and drill schedule artwork). Limited four-layer PCBs are also possible with the program. DASLL can be used in either batch-interactive or batch made of operation by technicians, draftsmen, designers, or engineers. Flexibility in being able to accomodate a diversity of trimline geometries and component shapes and placements was a goal in the design of the software. The system is also very flexible in its capability to deal with physical design rules. A principal objective for the system has been low-volume, quick turnaround response for low and medium density custom printed circuit boards.

I. INTRODUCTION

The DASLL (Design Automation System at Lawrence Livermore) is a system of programs to automatically generate the layout of printed circuit boards. The programming is done entirely in the LRLTRAN language which is an extended FORTRAN IV in use at LLL. The codes run on the Livermore Time Sharing Network on Control Data Corporation 7600 computer systems [3]. The goals in developing the DASLL system were to provide a software system aimed at providing a PCB service but flexible in that new services or applications could be easily added.

The motivation for developing the system was based on the PCB workload at Livermore. LLL designs 300 to 400 PCBs each year. The designs are custom boards used in prototype electronic systems and generally only a few (5 to 20) boards are produced for each design. Reworks are many (frequently 5 or more) and turnaround time is almost always very important. Many of the PCBs are layed out by outside contractors and when this occurs there is generally a feeling of "loss of control" by the designers: communication becomes more difficult, changes become more difficult, and schedules become more unpredictable.

Objectives were established early in the development of the PCB system. Based on computer aided systems elsewhere we felt confident of achieving a higher throughput per operator over traditional manual methods. We did not aim at handling 100% of all PCB requirements. Many special boards are required in various research programs, for example very small, very dense circular PCBs for satellites or highly layout-sensitive boards occuring in picosecond diagnostic measurements. These special boards are done manually. The majority of the boards (75% or greater) had relatively modest requirements on density, layout, geometry, etc. which we would tackle with the system. We knew that increased manufacturing precision would be a byproduct thus making certain aspects of manufacturing easier (for example the numerical controlled drilling operation as compared to the digitizing required for manual boards).

An important aspect of having an inhouse design automation system for support of R and D projects is local control and closer communication with the designer which is possible. Changes are easier and faster.

However, developing a design automation system in a R and D environment generated some unique requirements. Designers are, for the most part, free to choose hardware configurations, components, design rules, etc. as they see fit. Very few layout standards existed either for the engineers and designers or for the fabrication and inspection sections, only a commonly agreed upon sort of truce on what was and what wasn't acceptable. DA systems must work with a more fixed set of rules.

DASLL was designed to be flexible. For example, new PCB trimlines (outlines) could easily and quickly be entered in the system. New components and component outlines could be added to the data base in minutes. Design rules (conductor widths, spacing between conductors, feedthrough diameters, etc.) could be specified and used easily.

The above features and the conscientious use of software engineering practice in developing the system has yielded a very flexible, maintainable, and extendable DA system.

II. SYSTEM ORGANIZATION

Conceptually the organization of the DASLL PCB layout system is quite simple. Figure 1 reflects the organization in that the input is a description of the PC board type, components on the board, and the inter-connections to be made. After processing by the system several outputs are available: Reports, checkprint drawings, and NC tapes for machines.

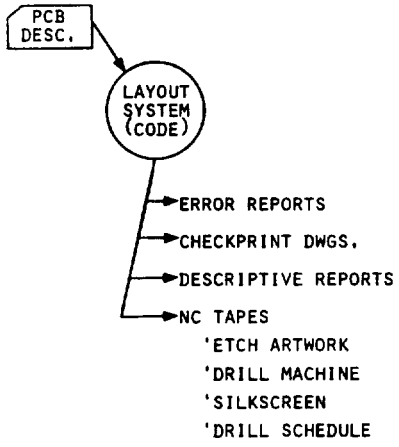

FIGURE 1. DASLL CONCEPTUALLY

The details of processing quickly become more complex. Figure 2 indicates a typical flow through the system. The modules preprocessor,placement, route, etc. are separately loaded subsystems which are run as individual programs but which share heavily in software modules.

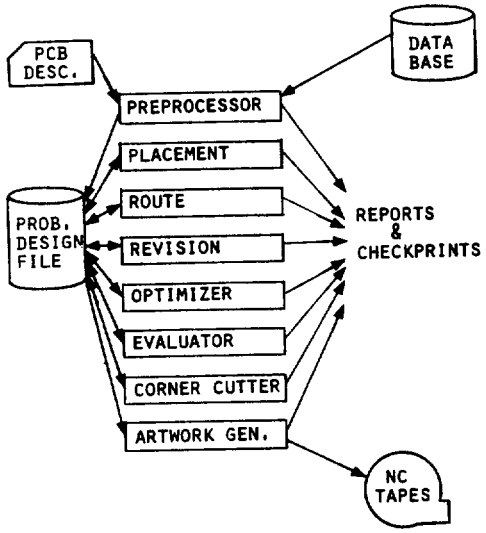

FIGURE 2. DASLL PROCESSING FLOW

The data base and problem design file are described in more detail in the rest of this section.

1. Data Base: The DASLL data base is intended to be the central repository for information related to design components, design description materials (PCB outlines, silkscreen patterns), and certain processing functions of the DA system. The data base consists of a collection of information, and the structures and conventions used to organize that information.

Insofar as practical, the data base software is intended to be independent of host system hardware and software. However in a few modules, specialized features are used, but these routines could easily be changed.

One of the most important features related to flexibility is the self-defining nature of the data base contents. Information describing the name, attributes, and location of each item is also stored in the data base file. The self-descriptive feature is implemented by defining a group of system data packets specifically for that task. All data in the data base and in the design file are in strict character format, arranged as standard 80-column hollerith card images. Each data packet is formed by one or more card images. Associated with each card image is a type code which is defined by a three-character mnemonic code in the first three columns of each card image of a packet.

Information in a data packet resides in a fixed field format on a card image of the packet. Skeleton data packets define (in position sensitive form) the starting column, the total character count, the data type, and other flags for each field in all other data packet types. The packet type and field information is used to access data from a program at run time and to specify fields to be altered in a data base update. Formats are generated to allow system I/O between the data base and design file.

A hierarchical data base structure is used. Each major application of the data base uses a section of the file which has a name and a level number two. Within each level are found subcollections of information each having a level name and a level number one greater than the containing level. For example, at one access time the DASLLDB level structure could look like:

```
                    I-> 2.SYSTYPE      I-> 3.PROMPTS
                    I-> 2.SYSDEFN      I-> 3.BOARD
                    I-> 2.SYSDOC       I-> 3.DEVICETYPE
1. DBSYSDATA->      I-> 2.USERS        I-> 3.PACKAGES
                    I-> 2.ROUTELIB->   I-> 3.WIRULES
                                       I-> 3.CONNECT
```

Within each level definition, one might find packets of information and/or definitions of subsidary levels.

2. Design File: The problem design file provides the communication of problem specific data among the subsystems. Each DASLL subsystem has an associated loader routine which must access the design file and load tables from it in a format used by the subsystem. The rationale for separating the loader from the body of the subsystem is that the many different subsystems each perform particular functions (e.g. routing or evaluation), all having common data needs but all not needing all of the data from the design file. Each subsystem is written to conform to a common design file, forcing informal consistency of the system.

All data in the design file are in strict character format, arranged in standard 80-column hollerith card images just as in the data base. The records are collected by type of information with each data packet in a fixed field format. Beginning each data type is a template record which defines the format for the following record group. These templates are defined in the data base.

Each DASLL subsystem proccessing adds to (or modifies) the problem design file until the entire processing sequence is completed. Design files for a particular layout are archived.

3. Communication: There are a large number of DASLL services available. To invoke these services a set of processing commands are used. Commands may be used in either batch or interactive mode. In both modes, the commands are executed in the order presented.

There are four categories of commands available, they are:

General System Commands,
Input-Output Commands,
Processing Commands, and
Database Maintenance Commands.

The processing commands, for example, include commands for revision, optimization, evaluation, and routing subsystems.

Under control of the DASLL supervisor, the user might, for example input data for a PCB layout. The input processor would extract the appropriate information from the data base and, with the appropriate command, write information into a problem design file. This information is passed to the router subsystem by a command, again processed by the supervisor, by instructing the router data loader routine to read from the design file. Figure 3 illustrates this flow of information. The input proccessor and the router are two DASLL subsystems.

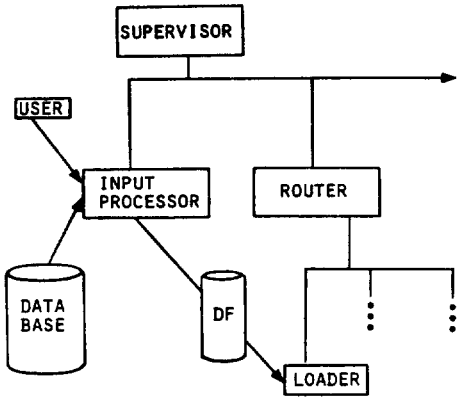

FIGURE 3. DASLL DATA TRANSFER/ COMMUNICATION FLOW

Internally, DASLL is controlled by a process, procedure, and sequence. A non-textural command exists for setting these three controls. The use of this command is useful in developing new features prior to installing a textural command. An example of a textural command is:

LIST COMMANDS

which lists all the available commands.

There are provisions in the DASLL supervisor system for 12 message level control variables. These variables control the amount and contents of one line process trace messages. Each variable influences the messages produced by a particular routine. It is the responsibility of each module designer to utilize the trace and conditional dump variables (also 12 in number) to produce trace and debug capabilities. The inclusion of control variables in common specifically for producing one line messages has been found useful for module testing and system integration testing. Concentrating the input-output to one level has also been useful for reducing system routines and thereby reducing overall size.

III. SYSTEM FEATURES

The DASLL PCB layout system offers several capabilities. In the following a few features of each subsystem will be highlighted.

1. Input Processor: Each PC board layout problem processed by DASLL has six major sections: 1) board identification, 2) board type, 3) location of components, 4) a list of signals connected to each component pin (signal pin list), 5) guidelines for layout (wire rules), and 6) design file format input. Although special input coding forms have been designed, the input is field free and extensive error checking is performed. Initial placement

of components is specified in the input and constraints on the mobility of the components can be specified: A component can be fixed in placement or restricted to a given X coordinate or a given Y coordinate. Components may also be given an affinity to each other which means that all members of a given class must be placed close to each other.

Although the preprocessor does extensive checking of the input, processing does continue in an effort to uncover as many errors as possible that may have been made in the input. The "trouble report" out of the preprocessor will indicate component overlaps, misspelled device or signal set names, improper device callouts, redundant device designations, and missing signal connections. Checkprint drawings showing the placement of components and a "rubber-band" model of all the connections specified in the input is also an output.

2. Placement: Although a subsystem of the DASLL system, the placement program has not been fully integrated into production use. The placement program is based on the use of unconnected sets or modules having no common signal sets. Random initial placement is used as the starting point for each iterative improvment cycle utilizing a choice of an unconnected set. The placement algorithm has not been fully evaluated.

3. Routers: Three router subsystems exist with the DASLL system. There are several ways the routing process can be viewed but for serial routers three major parts can be addressed:

1) Signal set decomposition,
2) Wire layout, and,
3) Clean-up.

Signal set decomposition can be further divided into two processes: 1) From-to generation, and 2) from-to ordering. All routers in the DASLL system use the same signal set decomposition; namely a minimum spanning tree signal set algorithm followed by a shortest-first ordering method. We have investigated other ordering methods and currently use a strategy which routes colinear X and colinear Y from-tos before the ordered from-to-list.

The three basic router techniques which have been implemented at Livermore are:

1) A channel router,
2) A line-probe router, and,
3) A maze-runner router.

We have done extensive studies of router evaluation [4, 5]. Most recently we have been using the rectangle-probe router with good success. Typically we have been achieving 98 to 100% completion rates for board densities of 1.1 square inches per equivalent integrated circuit and less dense. For densities from .5 to 1..1 we generally achieve 88 to 98% routing completion.

Because of the common design file, we can follow one router with another. For dense PCBs we have followed the rectangle-probe router with a slower Lee router before editing the layout. Normally our production sequence has used the probe router only.

4. Revision: Usually, unless the PCB is uncongested, automatic routing results in less than 100% routing. Three systems are available to edit revised boards under the DASLL system. When there are only a few changes, then working from the reports and checkprint drawings using revision commands is an effective and a fast method. When there are many unmades to be completed, then a digitizer simplifies the task of recording X-Y locations for revisions, additions, and deletions. A third method for editing the layout consists of using an autotrol interactive graphics system by transferring files to and from the autotrol system. This third method is not fully operational.

The revision subsystem reads in the design file and operates on the data structure representing the connection segment rectangles by adding segments or deleting segments for a given signal set or particular X-Y coordinate. On exit from the system the design file portion representing these segments is updated to reflect all changes.

5. Optimizer and Corner Cutter: Several reasons exist for having a route cleanup phase:

1) Remove undesired conductor configurations,

2) Reclaim board surface for futher router attempts,

3) Reduce production costs, and

4) Produce better looking boards.

The DASLL optimizer attempts to improve previously defined connection paths by eliminating unnecessary vias and adjusting the routing. A segment center line model is constructed for all the connections and attempts are made to add additional segments to create augmented paths. The augmented routes are evaluated on parameterized functions and a minimum spanning tree algorithm is used to select paths. Figure 4 illustrates some of the typical cleanup situations that the subsystem deals with effectively [6].

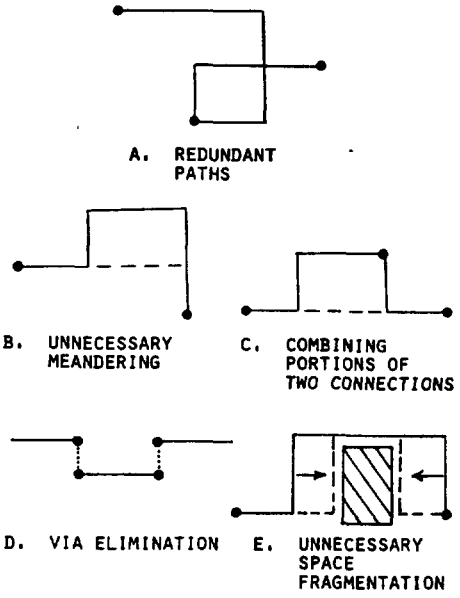

FIGURE 4. TYPICAL CLEANUP SITUATIONS

The corner cutter is almost a post processing phase of operating on the board layout. All right angle connection paths are "diagonlized" where possible. A board model very similiar to the optimizer model is constructed. The net result is a final layout which is not only suitable for production but more aesthetic. Figure 5 shows a portion of a PCB after corner cutting.

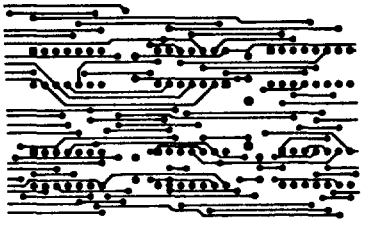

FIGURE 5. PCB LAYOUT AFTER CORNER CUTTING

6. Evaluator: The routing evaluator is an important component in the layout system in that it provides a means for analyzing the results produced by a router program and the effects of revision and optimization to see if the solution is valid [7]. The evaluator, using the system support routines, loads a description of the board and the layout. Next the evaluator accesses the data structure to ascertain:

1) The route barriers and board environment,

2) And for each distinct signal set:

   a) The pads which are to be electrically connected, and

   b) The links of conducting foil and feed-through vias produced by the router to make those connections.

When completed, the evaluator returns the values of three quality measures for the routed solution. The first is a route quality indicator which measures items like total path lenght compared to a hypothethical ideal, the number of feed-through vias, and the imbalance of foil area between board layers. The second is an electrical quality indicator which reflects the number of poor quality connections detected. The third measure is a count of the number of occurances of events which make the solution invalid and the number of links necessary to connect the incomplete routes.

7. Artwork Generator: There are currently four production outputs of the DASLL layout system:

1) Precision (1:1) film of the PCB etch artwork,

2) Precision (1:1) film of the silkscreen and drill schedule drawings,

3) Punched paper tapes for controlling the NC drills, and

4) Several reports including a parts list, drill size and locations, and electrical connection information.

IV. EXAMPLES AND PERFORMANCE CAPABILITIES

Based on over 75 printed circuit boards processed, the layout system can typically provide a turnaround in a production use of from four to seven days. In a few special cases a layout has gone from input to artwork in only one day.

Database contents include 11 PCB trimlines, over 130 component package outlines and almost 1000 device types.

As indicated earlier, the system routers deal effectively with board densities of about one square inch per integrated circuit. More dense and the revision stage gets difficult without an interactive graphics method of handling unmade connections.

The designs submitted to the DASLL system has been very mixed and varied. Figures 6 and 7 are superimposed (layer 1 and 2) etch artwork for a very simple layout of a type which occurs quite frequently (Fig. 6) and a medium density which also occurs often (Fig. 7).

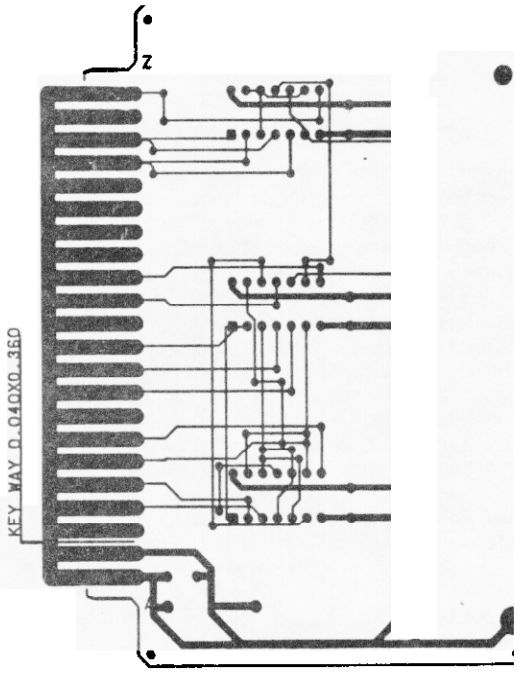

FIGURE 6. DASLL LAYOUT WITH BUS ROUTING

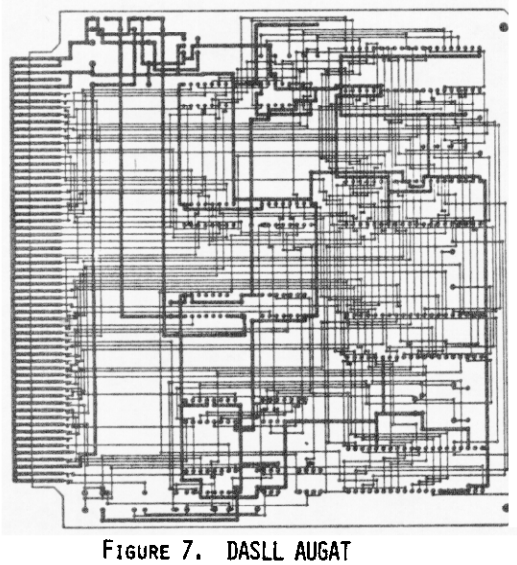

FIGURE 7. DASLL AUGAT PCB (7" x 7") LAYOUT

V. CONCULSIONS

The development of the LLL PCB layout system has led us to believe that cost-effective computer-based layout methods can be a realtiy. The successful move from development of a system to the production use of the system has several prerequisites. For example a minimum set of requirements for successful implementation are:

1) Input logging, review, selection, and assignment of all PCB jobs with a tightly controlled selection and assignment of those for automated processing,

2) Trained, capable, and motivated operators,

3) Close coordination amoung requestor, inspection, fabrication, and operator, and

4) Management commitment and support.

We have developed a number of forms to aid in the preparation, processing, and control of PC boards. These forms greatly help in 1) above. We have written training and user manuals [1, 2] for assistance with 2) above.

In benchmark evaluations it has appeared that an approximately 50% savings in time can be achieved by using automated layout process.

Work performed under the auspices of the U. S. Department of Energy by the Lawrence Livermore Laboratory under contract number W-7405-ENG-48.

REFERENCES

[1] Magnuson, W. G., Jr. and G. W. Willett, "DASLL User's Training and Procedures Manual," Lawrence Livermore Laboratory, Report UCID-18514, December 1979.

[2] Magnuson, W. G., Jr. and G. W. Willett, "DASLL User's Reference Manual," Lawrence Livermore Laboratory, (to be published).

[3] Magnuson, W. G., Jr. and G. W. Willett, "DASLL System Reference Manual," Lawrence Livermore Laboratory, Report UCID-18533, December 1979.

[4] Kelly, M. F., "Comparative Router Performance," Lawrence Livermore Laboratory, Report UCID-52319 (PhD Thesis), September 1977.

[5] Cage, W. G., and R. J. Smith, II, "A Rectange-Probe Router for Multilayer P.C. Boards," Proc. of the 14th Design Automation Conference, pp. 13-22, June 1977.

[6] Castleton, R. N., "A Procedure for Cleanup of Computer Routed Printed Circuit Boards," Conference Record 9th Asilomar Conference on Circuits, Systems and Computers, pp. 43-47, November 1975.

[7] Loomis, H. H., Jr., "A Computer Program to Evaluate the Quality of a PC Board Router," Conference Record 9th Annual Asilomar Conference on Circuits, Systems, and Computers," pp. 54-58, November 1975.